Creamos un proyecto de react

npx create-react-app myfirstredux

Probamos que nuestro proyecto de react corra bien. Si todo funciona OK, instalamos react-redux y redux

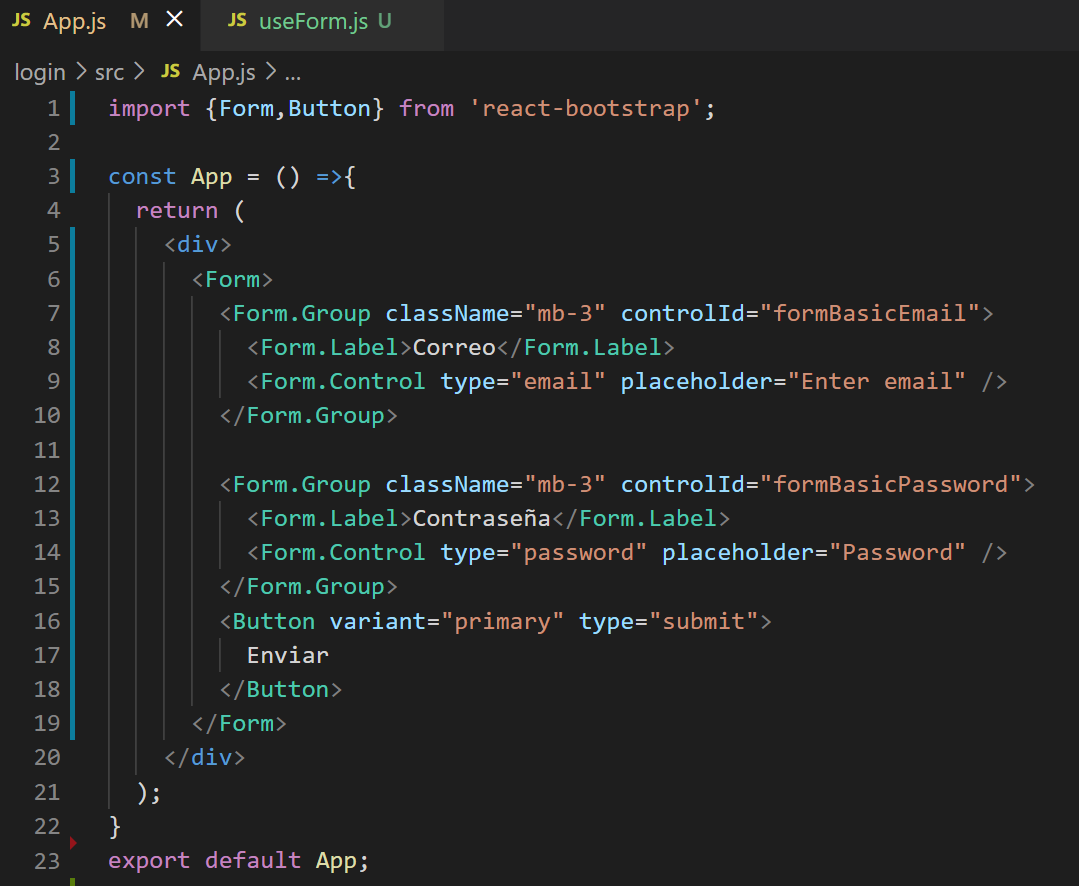
npm install react-redux redux

Verificamos su instalación en package.json

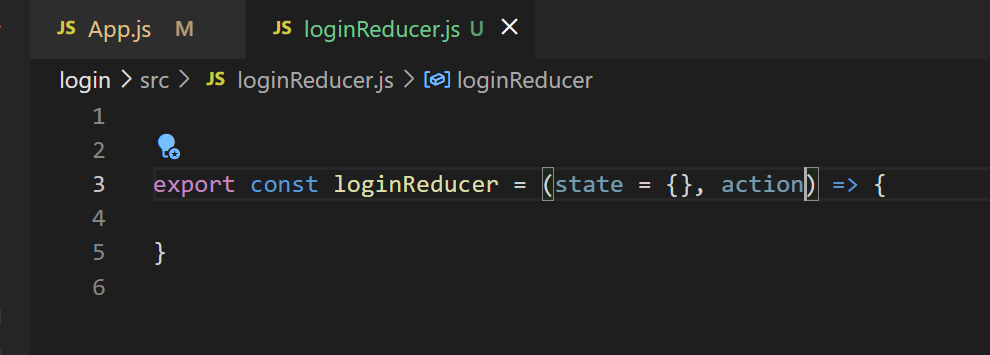
Instalamos react-bootstrap

npm install react-bootstrap bootstrap

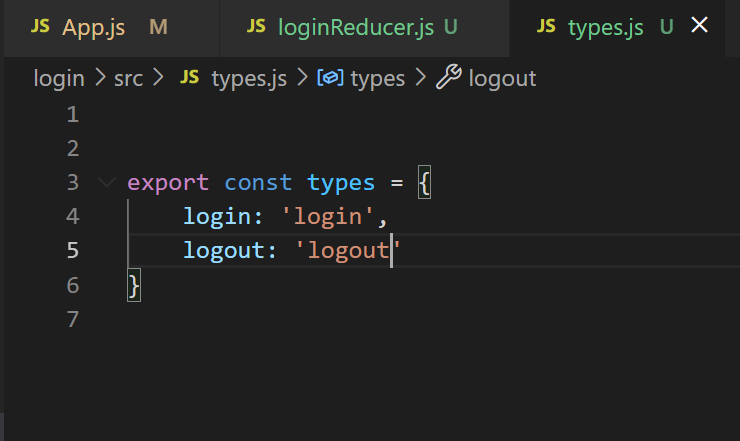
dentro del src nos quedamos con los archivos esenciales, modificamos nuestro archive App.js de la siguiente manera con la estructura del login



En src creamos un archivo llamado loginReducer, En este reducer vamos a tener el state vacío cuando no este autenticado



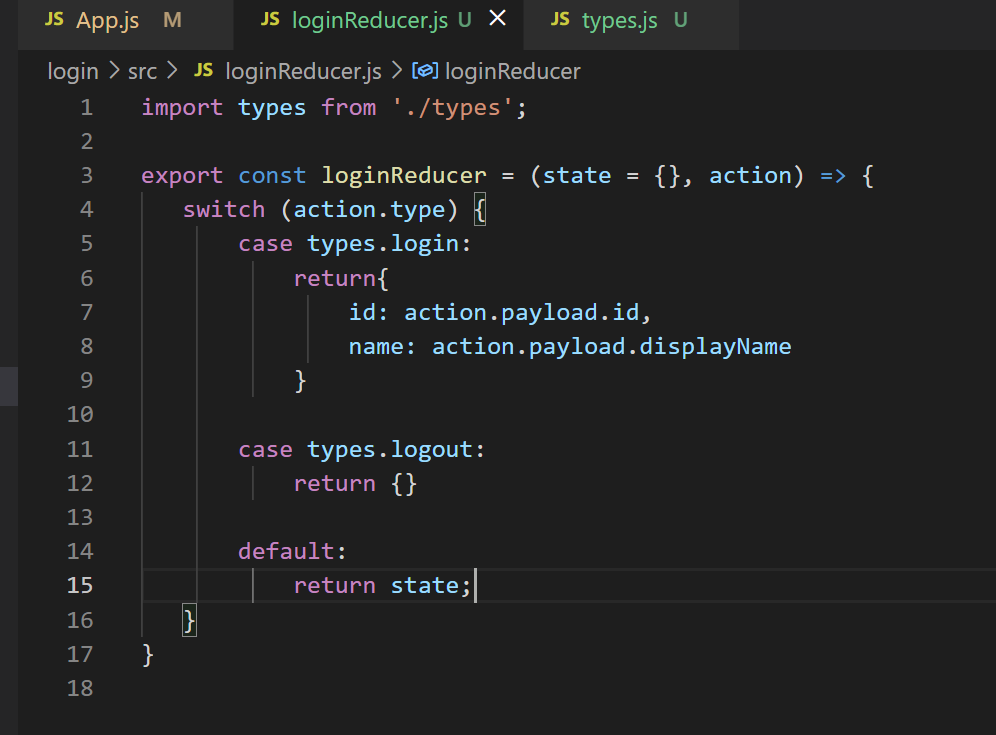
En src creamos un archivo que se llame types.js este contendrá nuestros tipos y creamos un objeto con ellos



Ahora en el reducer loginReducer vamos a configurar las acciones para cada uno de estos tipos

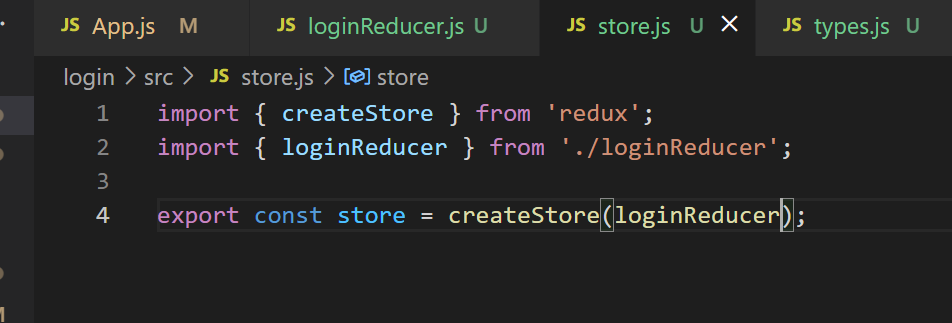
Importamos el archivo de types y estructuramos nuestro switch

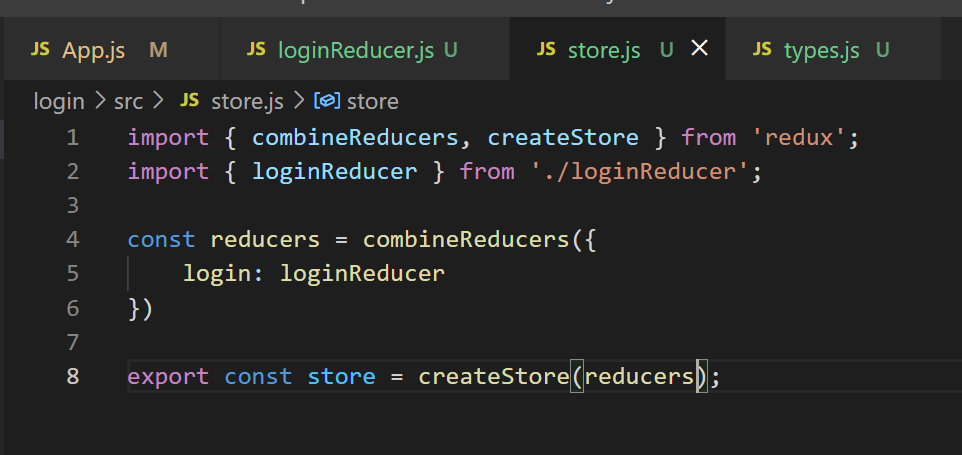
El logout sería restablecemos el objeto a un objeto vacío y el caso por defecto que va a retornar el mismo state



Vamos a crear el store (fuente única de la verdad) creamos un archivo llamado store.js

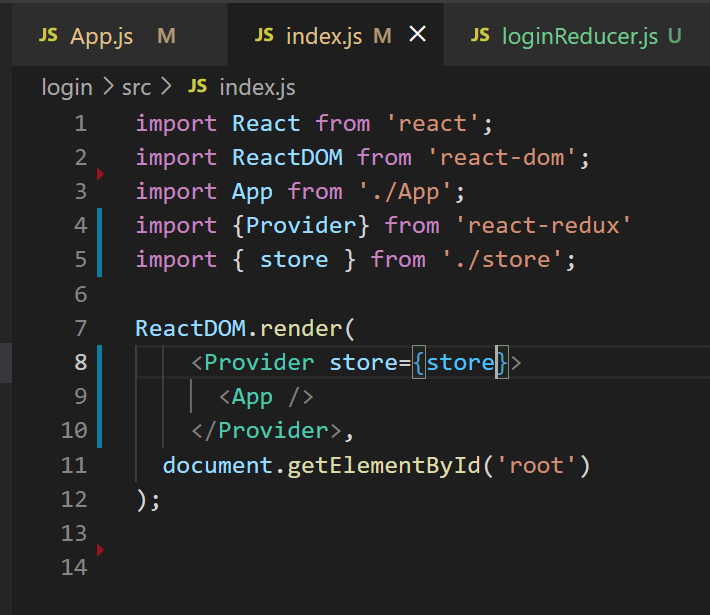
Podríamos enviar nuestro reducer dentro de la función, pero solo podemos enviar una, así que vamos a usar una nueva función que se llama combineReducers





Vamos a configurar Redux en nuestra aplicación, en este momento el archivo login y store están flotando, aún no los estamos usando en react para nada.

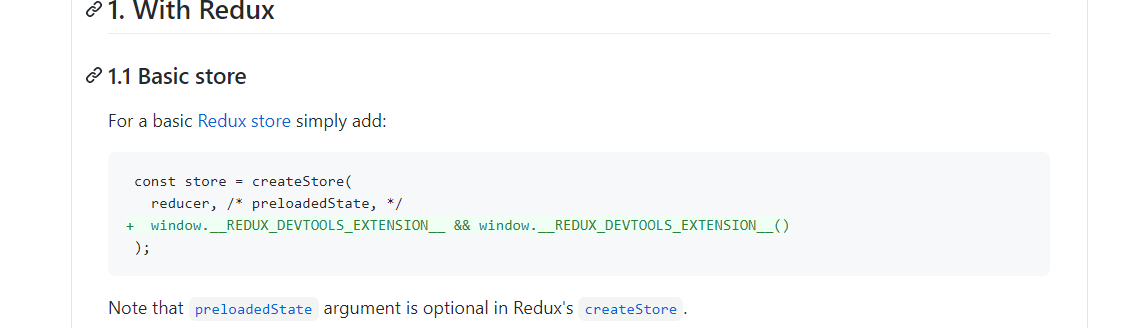
Vamos al archivo index.js y vamos a envolver el componente App.js en el El provider va a tener la información que en este caso sería el store

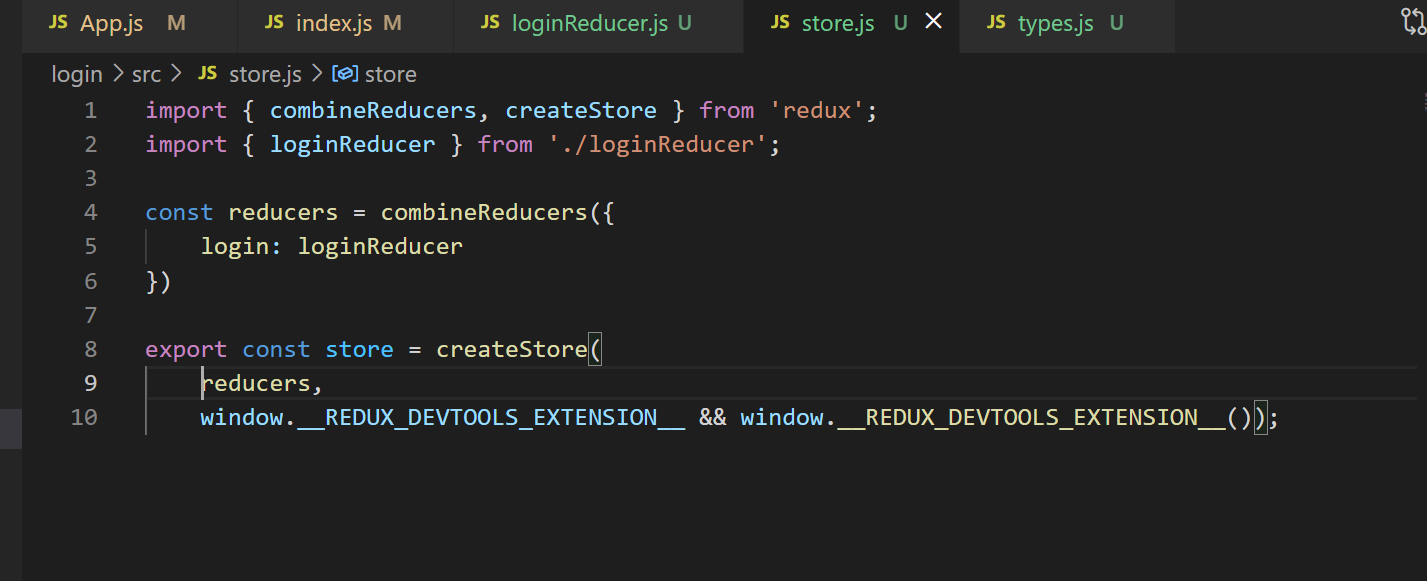


DEV TOOLS REDUX

Para configurar el store vamos a ingresar a la pagina <https://github.com/zalmoxisus/redux-devtools-extension#usage>

Vamos a copiar la línea verde y la copiamos dentro del créate store

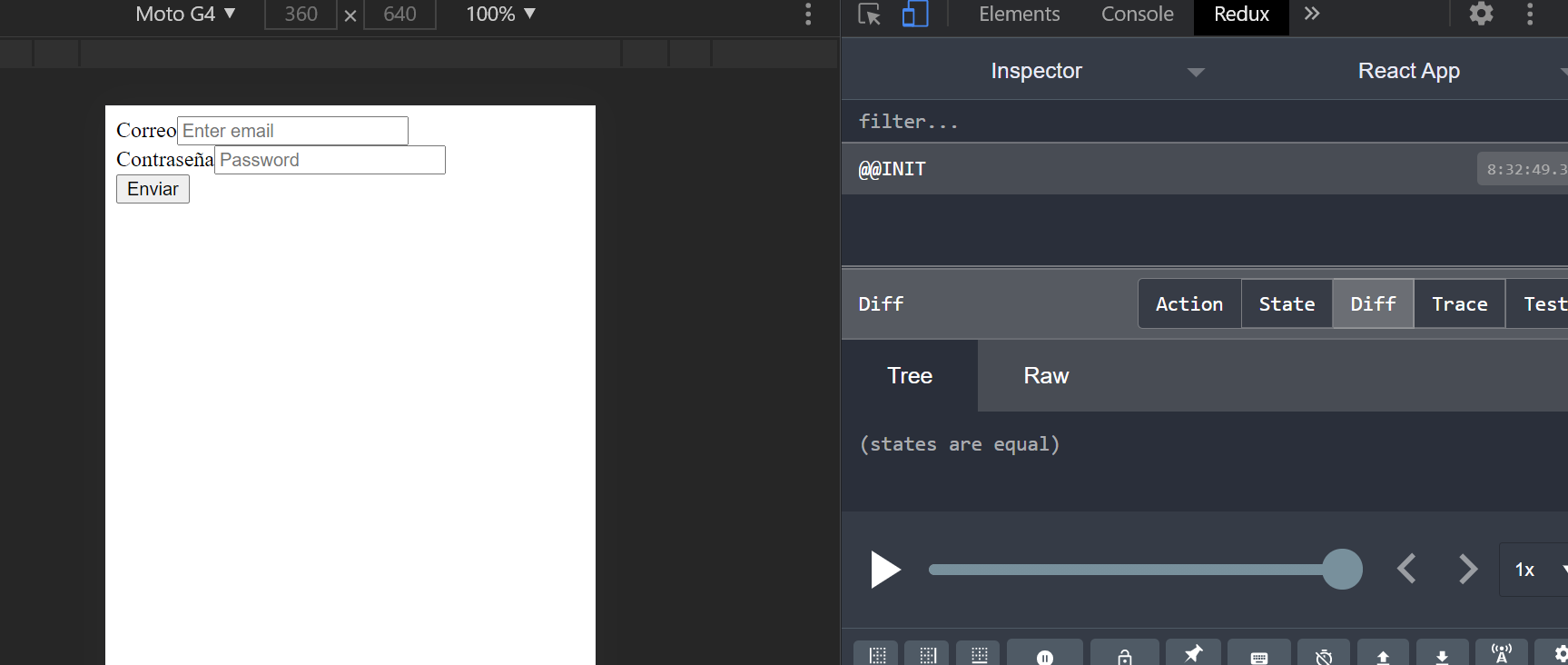




Si no les aparece instalar esta extensión

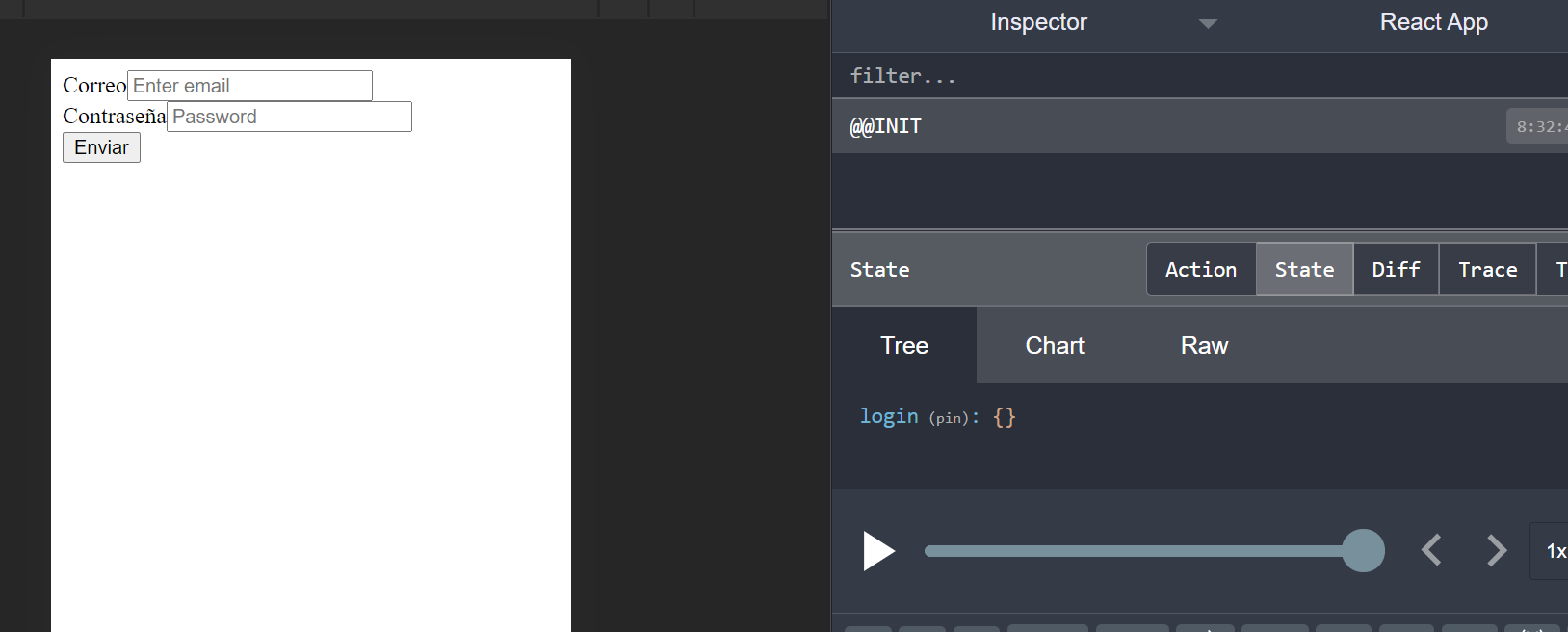
Instalar la extensión <https://chrome.google.com/webstore/detail/redux-devtools/lmhkpmbekcpmknklioeibfkpmmfibljd/related?hl=es>

cerrar el navegador y volver a correr el proyecto

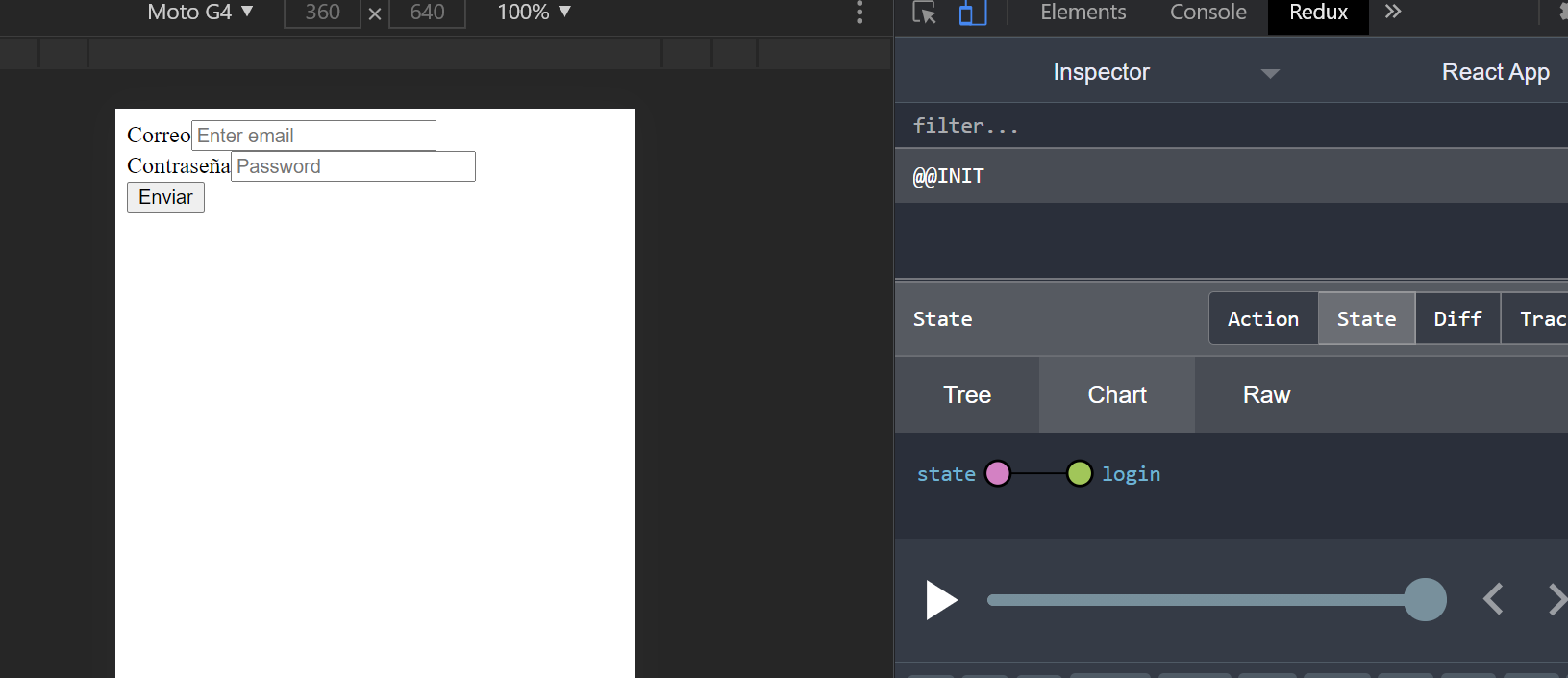


En este devtools de redux

Vemos el state, acá podemos ver nuestro state



Y en chart lo único que tenemos es el state y nuestro objeto ath



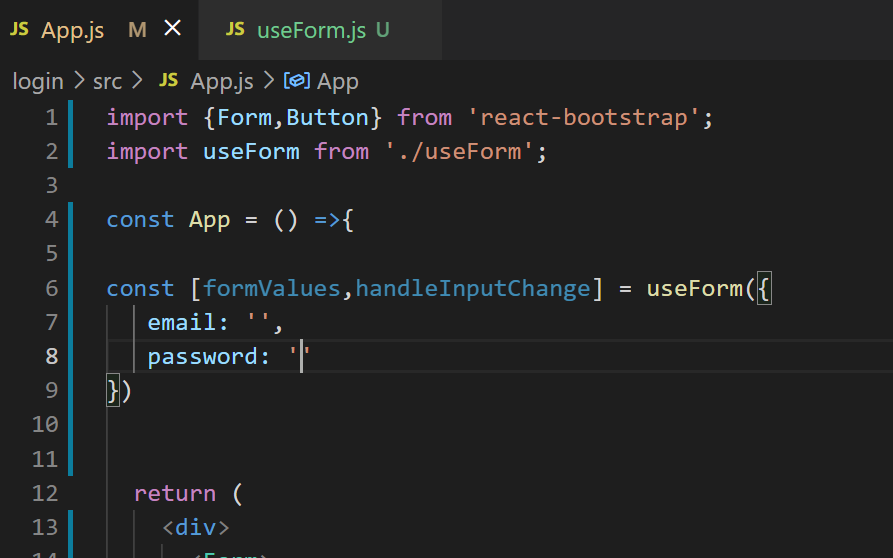
Vamos a hacer la siguiente prueba. Nos vamos para el archivo loginReducer

https://raw.githubusercontent.com/Klerith/mis-custom-hook/master/useForm/useForm.js vamos a usar un hook que nos crea un objeto

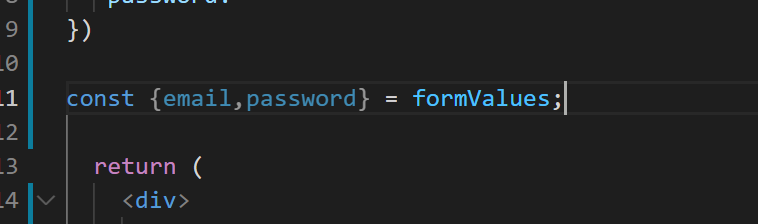
creamos en src un archivo llamado useForm.js ingresamos al link y pegamos el código para manejar formularios

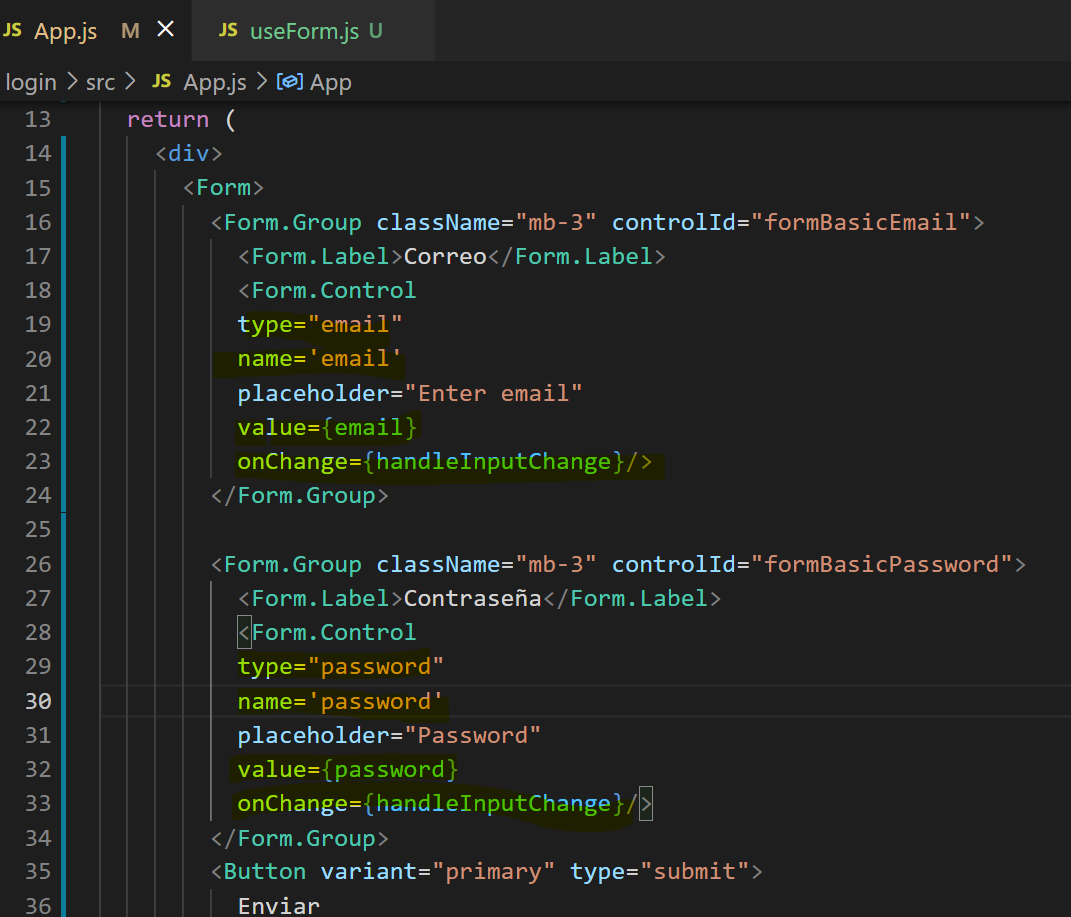
Vamos a tomar ese useForm y lo vamos a implementar en el App.js

El estado inicial lo voy a definir directamente en el useForm

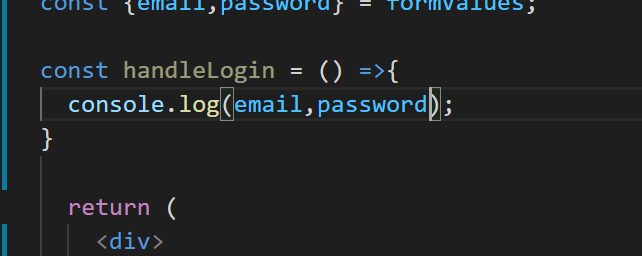


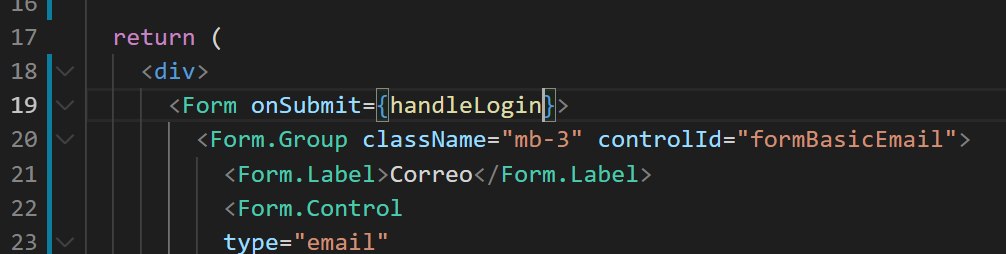
Desestructuramos el form values y los asignamos en sus respectivos form-control y en onchang la función handle



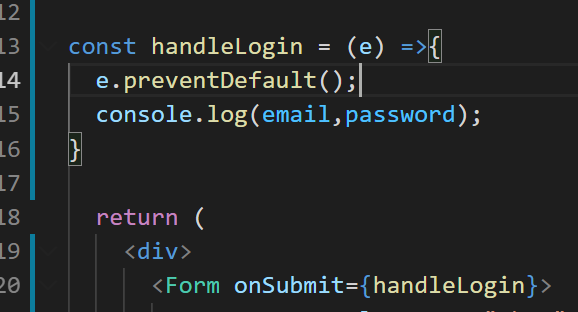


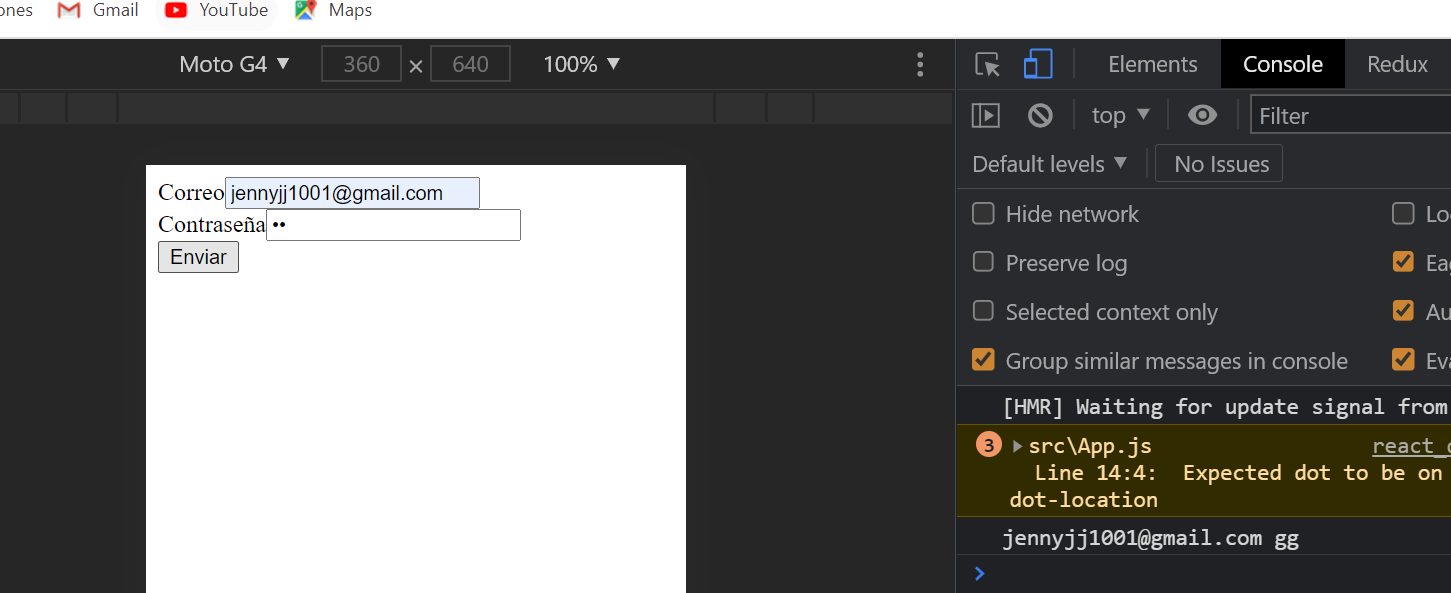
Vamos a manejar el submit del formulario





Recibimos el evento y evitamos la propagación del formulario



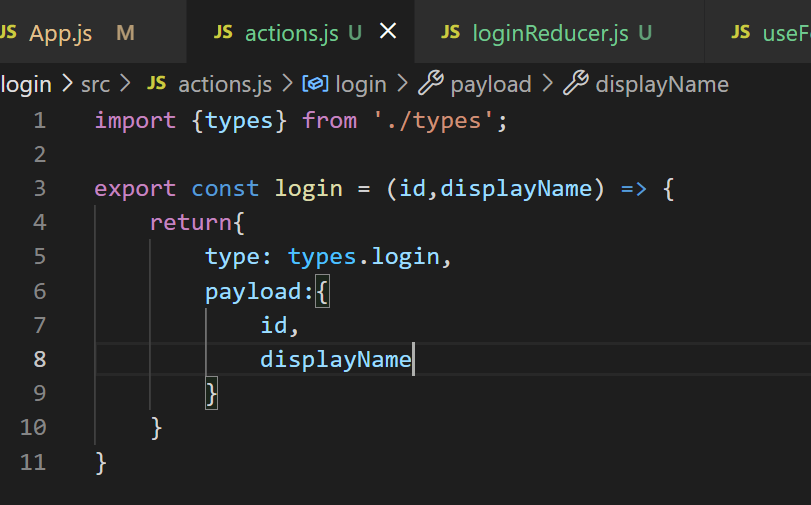


Hacemos un dispatch

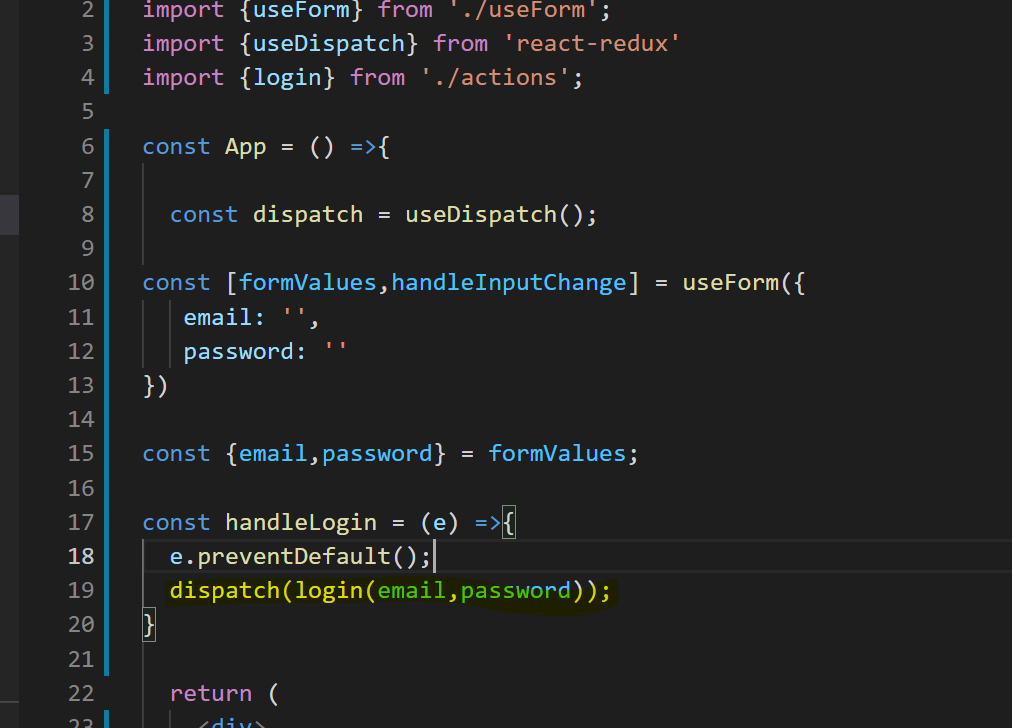
Vamos a crear una acción para hacer un dispatch

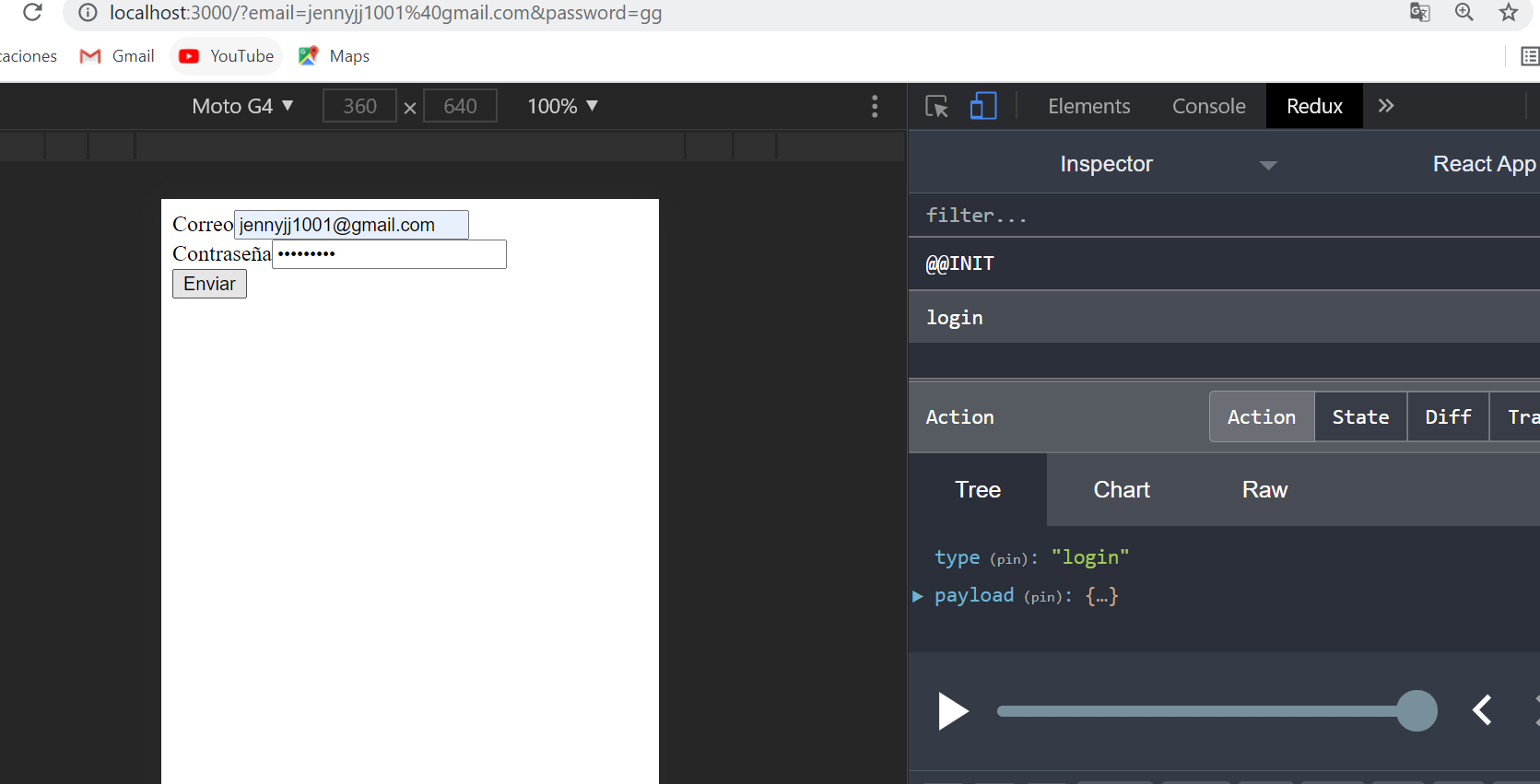
Creamos un archivo actions.js

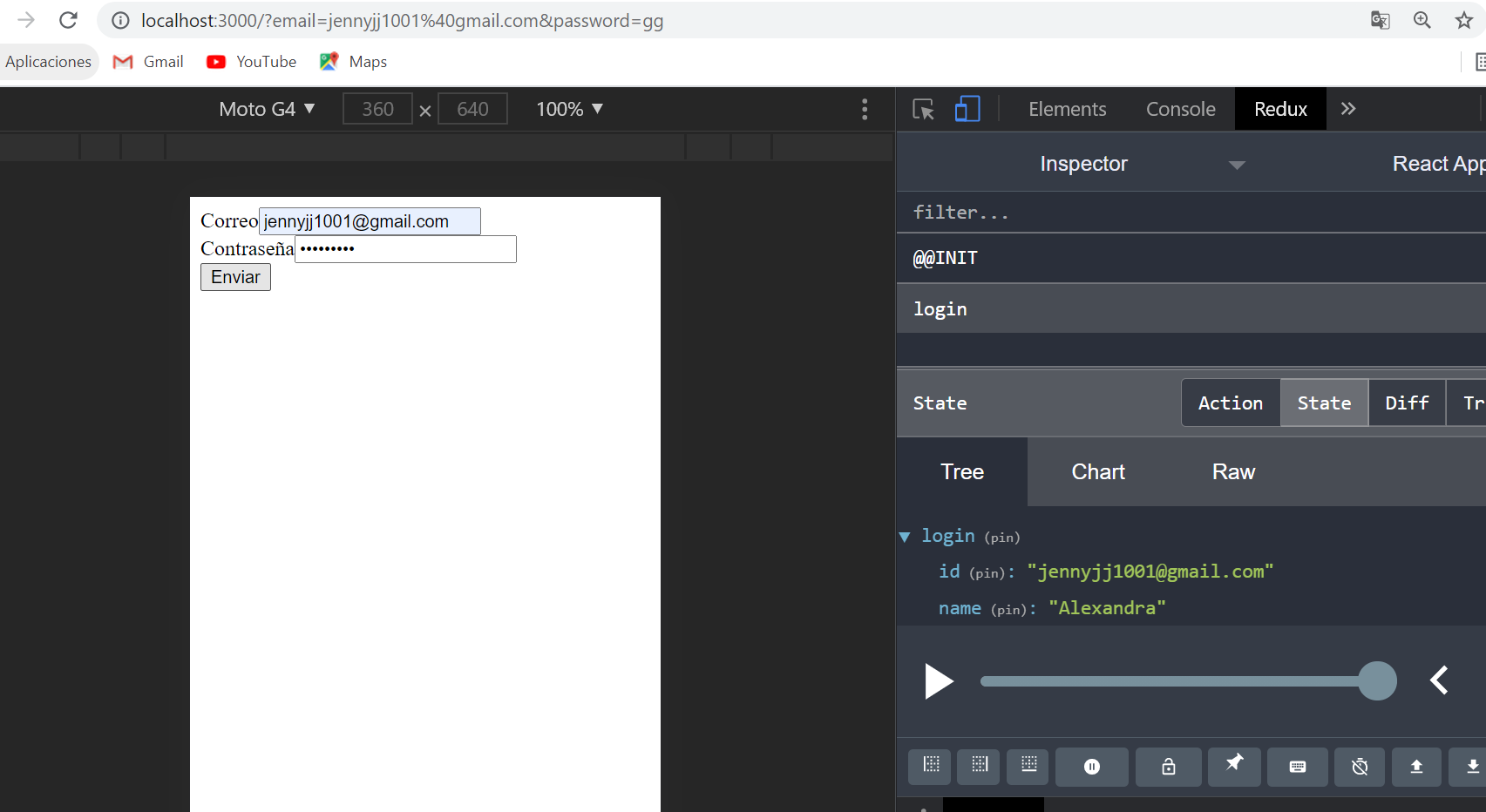
actions y vamos a agrupar cada una de las acciones en un respectivo archivo



Archivo App.js







Actividad:

Teniendo en cuenta la explicación de la clase. Realizar una aplicación para un formulario de registro que contenga los siguientes campos

Nombre

Apellido

Teléfono

Observaciones

Incluir hooks

Redux

Importamos useDispatch

